

BUILDING END-TO-END SERVERLESS WEB APPLICATION

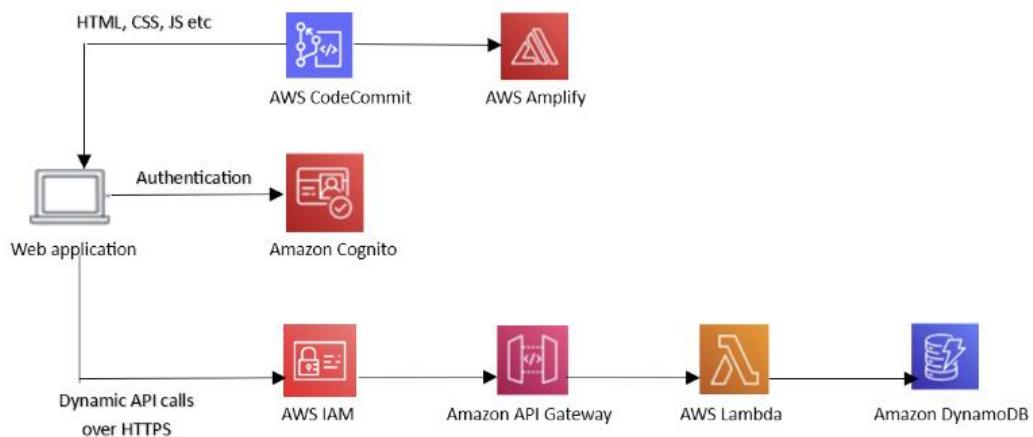
Name: D. Lavanya Chowdary

Id: 2100032412

SERVICES USED:

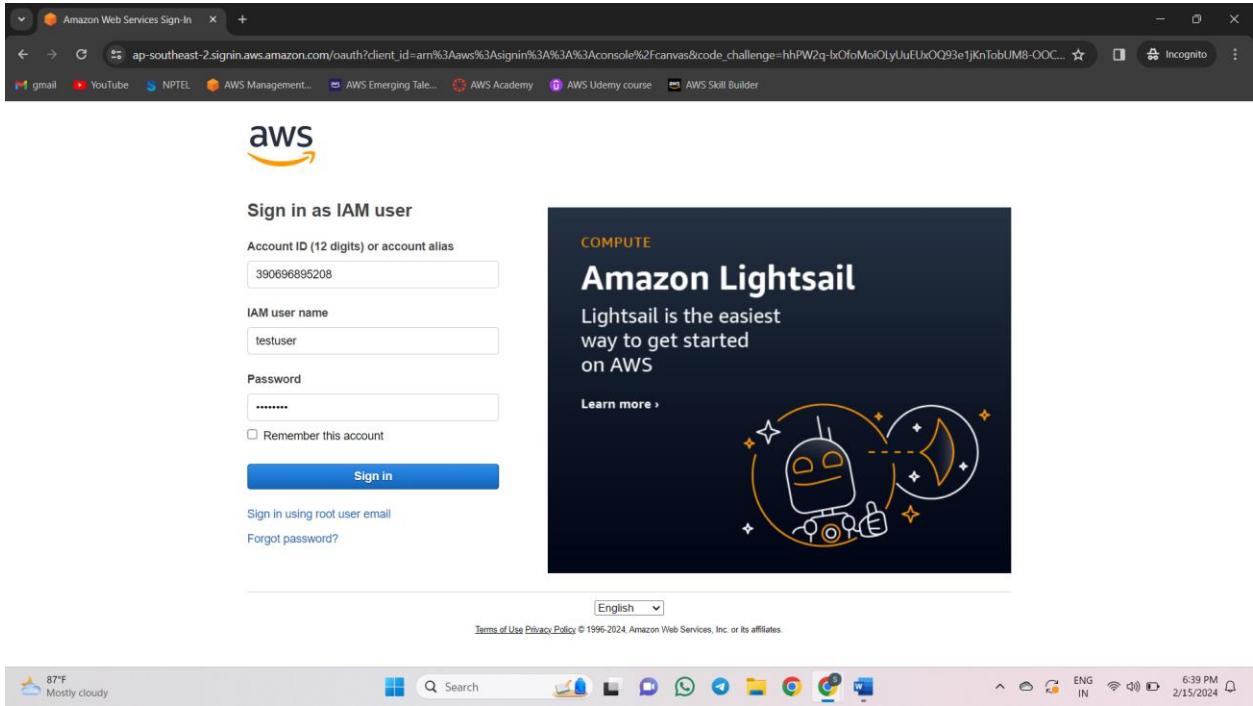
1. AWS CodeCommit
2. AWS Amplify
3. Amazon Cognito
4. AWS IAM
5. Amazon API Gateway
6. AWS Lambda
7. Amazon DynamoDB

SERVICE ARCHITECTURE:



PROCEDURE:

Create IAM User named “testuser” and open in incognito tab with given console.
Open AWS IAM user account using the credentials



The screenshot shows the AWS Console Home page. On the left, under 'Recently visited', the 'IAM' service is listed. On the right, the 'Applications' section is displayed, showing a table with columns for Name, Description, Region, and Originating account. A message indicates 'No applications' and provides a link to 'Create application'. Below this, there's a link to 'Go to myApplications'. The top navigation bar includes links for Gmail, YouTube, NPTEL, AWS Management Console, AWS Emerging Tech, AWS Academy, AWS Udemy course, and AWS Skill Builder. The status bar at the bottom shows the date and time as 6:39 PM on 2/15/2024.

Create a repository in CodeCommit named “wildrydes-site”

The screenshot shows the 'Create repository' page in AWS CodeCommit. In the 'Repository settings' section, the 'Repository name' field is filled with 'wildrydes-site'. There is an optional 'Description' field and a 'Tags' section with an 'Add tag' button. Below these, there are sections for 'Additional configuration' (AWS KMS key) and 'Enable Amazon CodeGuru Reviewer for Java and Python' (optional). The top navigation bar includes links for Gmail, YouTube, NPTEL, AWS Management Console, AWS Emerging Tech, AWS Academy, AWS Udemy course, and AWS Skill Builder. The status bar at the bottom shows the date and time as 6:44 PM on 2/15/2024.

The screenshot shows a browser window for AWS CodeCommit. The URL is us-west-2.console.aws.amazon.com/codesuite/codecommit/repositories/wildrydes-site/setup?region=us-west-2. A green success banner at the top says "Repository successfully created". Below it, the repository name "wildrydes-site" is displayed. On the left, a sidebar menu for "CodeCommit" includes options like "Source", "Code", "Artifacts", "Build", "Deploy", "Pipeline", and "Settings". The main content area shows "Connection steps" with tabs for "HTTPS", "SSH", and "HTTPS (GRC)". It provides instructions for prerequisites, git credentials, and cloning the repository. A "Clone URL" button is visible. The bottom of the screen shows the AWS navigation bar and system status.

Add a policy to your IAM user “testuser”

The screenshot shows the "Add permissions" wizard in the AWS IAM console. The URL is us-east-1.console.aws.amazon.com/iamv2/home?region=us-west-2#/users/details/testuser/add-permissions. The "Step 1: Add permissions" section is active. It shows three options: "Add user to group" (radio button), "Copy permissions" (radio button), and "Attach policies directly" (radio button, highlighted). The "Permissions policies" section lists three policies: "AWSCodeCommitFullAccess" (unchecked), "AWSCodeCommitPowerUser" (checked), and "AWSCodeCommitReadOnly" (unchecked). The checked policy is highlighted. The bottom of the screen shows the AWS navigation bar and system status.

Go to security credentials and generate credentials

The screenshot shows the AWS IAM console with the 'Generate credentials' dialog box open. The dialog box displays a green message: 'Your new credentials are available.' It also contains instructions: 'Save your user name and password or download the credentials file.' Below this, it says, 'This is the only time you can view the password or download it. You cannot recover it later. However, you can reset your password at any time.' A note states, 'You can use these credentials when connecting from your local computer, or from tools that require a static user name and password.' The dialog box includes fields for 'User name' (testuser-at-390696895208) and 'Password' (redacted). At the bottom are 'Download credentials' and 'Close' buttons.

Clone the url in CodeCommit

The screenshot shows the AWS CodeCommit console for the 'wildrydes-site' repository. On the left, a sidebar menu for 'CodeCommit' is visible with options like 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The main area displays 'Connection steps' with tabs for 'HTTPS' (selected), 'SSH', and 'HTTPS (GRC)'. It provides instructions for 'Step 1: Prerequisites', 'Step 2: Git credentials', and 'Step 3: Clone the repository'. A command line is shown: 'git clone https://git-codecommit.us-west-2.amazonaws.com/v1/repos/wildrydes-site'. A 'Copy' button is next to the command. At the top right, there are buttons for 'Clone URL', 'Clone HTTPS', 'Clone SSH', and 'Clone HTTPS (GRC)'. The status bar at the bottom indicates 'CloudShell' and 'Feedback'.

Open cloud shell and clone an empty repository

The screenshot shows a browser window with the AWS Developer Tools interface. A modal window titled 'Copied' displays the URL <https://git-codecommit.us-west-2.amazonaws.com/v1/repos/wildrydes-site>. Below this, the 'wildrydes-site' repository page is visible, showing a 'Connection steps' section. An AWS CloudShell window is open, showing the command line:

```
[cloudshell-user@ip-10-130-33-6 ~]$ git clone https://git-codecommit.us-west-2.amazonaws.com/v1/repos/wildrydes-site
Cloning into 'wildrydes-site'...
Username for 'https://git-codecommit.us-west-2.amazonaws.com': testuser-at-390696895208
Password for 'https://testuser-at-390696895208@git-codecommit.us-west-2.amazonaws.com':
warning: You appear to have cloned an empty repository.
[cloudshell-user@ip-10-130-33-6 ~]$ ls
[cloudshell-user@ip-10-130-33-6 ~]$ cd wildrydes-site/
[cloudshell-user@ip-10-130-33-6 wildrydes-site]$ ls
[cloudshell-user@ip-10-130-33-6 wildrydes-site]$
```

The CloudShell window has tabs for 'CloudShell' and 'Feedback'. The taskbar at the bottom shows various application icons.

Copy project code from s3 bucket

The screenshot shows a browser window with the AWS Developer Tools interface. A modal window titled 'Copied' displays the URL <https://git-codecommit.us-west-2.amazonaws.com/v1/repos/wildrydes-site>. Below this, the 'wildrydes-site' repository page is visible, showing a 'Connection steps' section. An AWS CloudShell window is open, showing the command line:

```
[cloudshell-user@ip-10-130-33-6 ~]$ git clone https://git-codecommit.us-west-2.amazonaws.com/v1/repos/wildrydes-site
Cloning into 'wildrydes-site'...
Username for 'https://git-codecommit.us-west-2.amazonaws.com': testuser-at-390696895208
Password for 'https://testuser-at-390696895208@git-codecommit.us-west-2.amazonaws.com':
warning: You appear to have cloned an empty repository.
[cloudshell-user@ip-10-130-33-6 ~]$ ls
[cloudshell-user@ip-10-130-33-6 ~]$ cd wildrydes-site/
[cloudshell-user@ip-10-130-33-6 wildrydes-site]$ ls
[cloudshell-user@ip-10-130-33-6 wildrydes-site]$ aws s3 cp s3://ttt-wildrydes/wildrydes-site ./ --recursive
```

The CloudShell window has tabs for 'CloudShell' and 'Feedback'. The taskbar at the bottom shows various application icons.

The screenshot shows a browser window with two tabs: "wildrydes-site | AWS Developer" and "testuser | IAM | Global". The main content area displays the AWS CodeCommit repository "wildrydes-site". A modal window titled "AWS CloudShell" is open, showing a terminal session. The terminal output is as follows:

```
download: $1://t/t/wildrydes/wildrydes-site/verify.html to ./verify.html
download: $1://t/t/wildrydes/wildrydes-site/js/vendor/amazon-cognito-identity.min.js to js/vendor/amazon-cognito-identity.min.js
download: $1://t/t/wildrydes/wildrydes-site/js/vendor/bootstrap.min.js to js/vendor/bootstrap.min.js
[ccloudshell-user@ip-10-130-33-6 wildrydes-site]$ ls
index.html  investors.html  js  register.html  ride.html  robots.txt  signin.html  unicorns.html  verify.html
[ccloudshell-user@ip-10-130-33-6 wildrydes-site]$ git add .
[ccloudshell-user@ip-10-130-33-6 wildrydes-site]$ git commit -m "Initial commit"
Author identity unknown
*** Please tell me who you are.
Run
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
to set your account's default identity.
Omit --global to set the identity only in this repository.
fatal: empty ident.name (for <ccloudshell-user@ip-10-130-33-6.us-west-2.compute.internal>) not allowed
[ccloudshell-user@ip-10-130-33-6 wildrydes-site]$
```

The CloudShell interface includes a "Clone URL" button and an "Actions" dropdown menu. The status bar at the bottom shows "CloudShell Feedback", the date "2/15/2024", and the time "7:02 PM".

This screenshot is identical to the one above, showing the same AWS CodeCommit repository and CloudShell terminal session. The terminal output is the same, indicating a failed attempt to commit due to an empty identity name.

Check the files copied from s3

The screenshot shows the AWS CodeCommit interface. On the left, a sidebar menu for 'CodeCommit' is open, showing options like 'Source', 'Getting started', 'Repositories', 'Code', 'Approval rule templates', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The main content area displays the 'wildrydes-site' repository. It shows a list of files under the 'Info' tab, including 'css', 'fonts', 'images', 'js', 'apply.html', 'faq.html', 'favicon.ico', 'index.html', 'investors.html', and 'register.html'. There are buttons for 'Notify' (dropdown), 'master' (dropdown), 'Create pull request', and 'Clone URL'.

Go to Amplify and host web app

The screenshot shows the AWS Amplify interface. On the left, a sidebar menu for 'AWS Amplify' is open, showing 'All apps', 'Manage Sandboxes (New)', 'Documentation', and 'Support'. The main content area features a 'Get started' section with two main sections: 'Amplify Studio' and 'Amplify Hosting'. 'Amplify Studio' includes an icon of a cloud with a plus sign, a 'Build an app' description, and a 'Get started' button. 'Amplify Hosting' includes an icon of a globe and a laptop, a 'Host your web app' description, and a 'Get started' button. At the bottom, there is a note about connecting existing AWS resources with Amplify Libraries and a link to the documentation.

wildrydes-site | AWS Developer testuser | IAM | Global AWS Amplify | us-west-2

us-west-2.console.aws.amazon.com/amplify/home?region=us-west-2#/create

Gmail YouTube NPTEL AWS Management... AWS Emerging Tale... AWS Academy AWS Udemy course AWS Skill Builder

AWS Services Search [Alt+S]

AWS Amplify

All apps Manage Sandboxes (New)

Documentation Support

Get started with Amplify Hosting

Amplify Hosting is a fully managed hosting service for web apps. Connect your repository to build, deploy, and host your web app.

From your existing code

Connect your source code from a Git repository or upload files to host a web app in minutes.

GitHub 

Bitbucket 

GitLab 

AWS CodeCommit 

Deploy without Git provider 

Amplify Hosting requires read-only access to your repository.

Continue

AWS CloudShell CloudShell Feedback 87°F Mostly cloudy Search Actions Privacy Cookie preferences © 2024, Amazon Web Services, Inc. or its affiliates. Terms 7:16 PM 2/15/2024

wildrydes-site | AWS Developer testuser | IAM | Global AWS Amplify | us-west-2

us-west-2.console.aws.amazon.com/amplify/home?region=us-west-2#/create

Gmail YouTube NPTEL AWS Management... AWS Emerging Tale... AWS Academy AWS Udemy course AWS Skill Builder

AWS Services Search [Alt+S]

AWS Amplify

All apps Manage Sandboxes (New)

Documentation Support

Add repository branch

AWS CodeCommit

✓ AWS CodeCommit authorization was successful.

Repository service provider 

Recently updated repositories
If you don't see your repository below, please push a commit and then click the refresh button.
wildrydes-site

Branch
Select a branch from your repository:

Connecting a monorepo? Pick a folder.

Cancel Previous Next

AWS CloudShell CloudShell Feedback 87°F Mostly cloudy Search Actions Privacy Cookie preferences © 2024, Amazon Web Services, Inc. or its affiliates. Terms 7:16 PM 2/15/2024

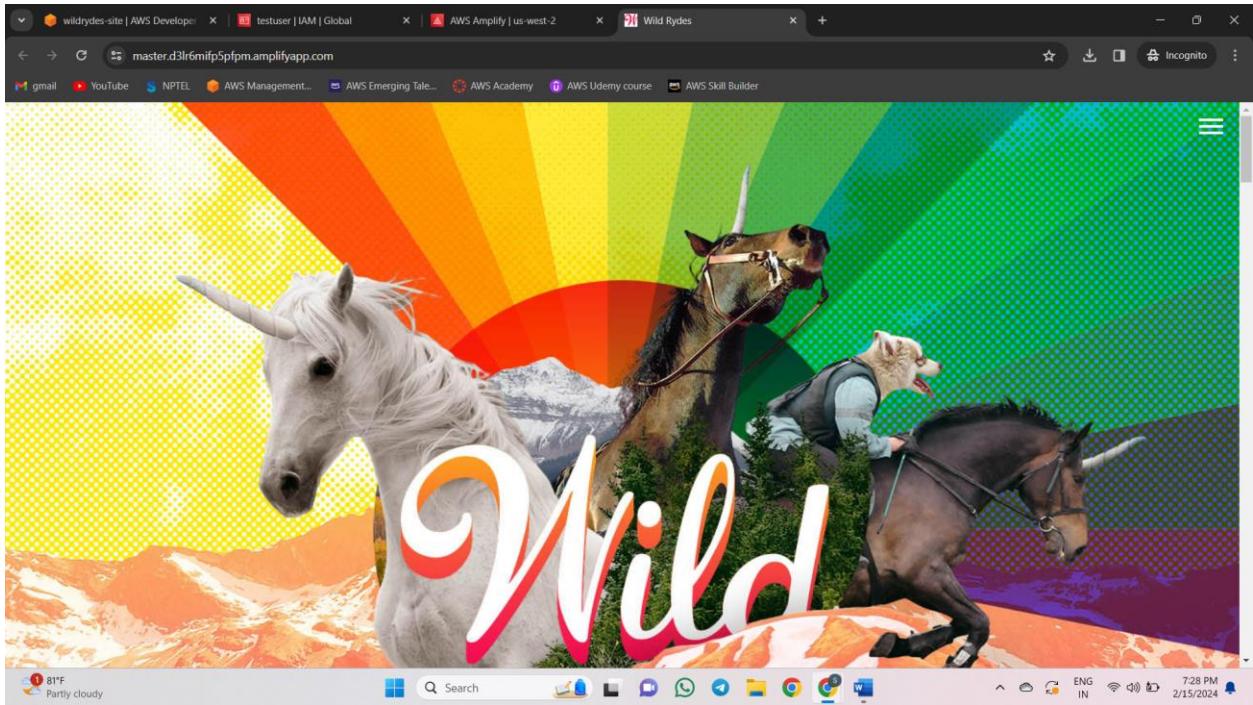
The screenshot shows the AWS Amplify Step 3: Review screen. The 'App name' field contains 'wildrydes-site'. Under 'Build and test settings', a code editor displays a build command:

```
version: 1
frontend:
  phases:
    # IMPORTANT - Please verify your build commands
    build:
      commands: []
    artifacts:
      # IMPORTANT - Please verify your build output directory
      baseDirectory: /
      files:
        - "*/**/*"
      cache:
      paths: []
```

Below the code editor are 'Download' and 'Edit' buttons. A checked checkbox allows automatic deployment. Under 'Server-Side Rendering (SSR) deployment', there is a checked checkbox for enabling SSR app logs.

The screenshot shows the AWS Amplify homepage for the 'wildrydes-site' app. The left sidebar includes 'All apps' and 'wildrydes-site' under 'App settings'. The main content area displays a 'Learn how to get the most out of Amplify Hosting' banner and a 'Hosting environments' tab. It lists a single environment named 'master' with a status bar showing 'Continuous deploys set up (Edit)'. Below it is a preview card for 'https://master...amplifyapp.com' and a deployment pipeline diagram with three stages: Provision, Build, and Deploy, all marked as successful. The pipeline details show the last deployment at 2/15/2024, 7:18:31 PM, the last commit message 'This is an autogenerated message | Auto-build | AWS CodeCommit - master', and that Previews are disabled.

Open the url



Edit the index.html in repository and check in Amplify

```
<!DOCTYPE html>
<html class="no-js" lang="en">
<head>
<meta charset="utf-8">
<meta name="description" content="">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Wild Rydes</title>
<link rel="stylesheet" href="css/font.css">
<link rel="stylesheet" href="css/main.css">
<script src="js/vendor/modernizr.js"></script>
</head>
<body class="page-home">
<header class="site-header">
<h1>Wild Rydes</h1>
<nav class="site-nav">
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="about.html">Meet Our Unicorns</a></li>
<li><a href="investors.html">Investors & Board of Directors</a></li>
<li><a href="faq.html">FAQ</a></li>
<li><a href="apply.html">Apply</a></li>
</ul>

```

The screenshot shows the AWS CloudShell interface with the AWS Lambda function configuration page open. The Lambda function name is 'wildrydes-site'. The configuration tab is selected, showing the following details:

- Function name:** wildrydes-site
- Runtime:** Node.js 18.x
- Memory:** 128 MB
- Timeout:** 300 seconds
- Environment:** No environment variables are defined.
- Code:** The code editor shows the function's source code in Node.js.
- Logs:** The logs tab shows the execution history of the function.
- Deployment:** The deployment tab shows the deployment history and triggers.
- Metrics:** The metrics tab shows the monitoring data for the function.
- CloudWatch Metrics:** The metrics tab also shows CloudWatch Metrics settings.

Create user pool in Cognito for authentication

The screenshot shows the AWS CloudShell interface with the AWS Cognito User Pools creation page open. The URL is <https://us-west-2.console.aws.amazon.com/cognito/v2/idp/user-pools/create?region=us-west-2>. The process is at Step 1: Configure sign-in experience.

Configure sign-in experience

Your app users can sign in to your user pool with a user name and password, or sign in with a third-party identity provider.

Authentication providers

Configure the providers that are available to users when they sign in.

Provider types

Choose whether users will sign in to your Cognito user pool, a federated identity provider, or both. Amazon Cognito has different pricing for federated users and user pool users. Learn more about pricing [\[?\]](#)

Cognito user pool
Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.

Federated identity providers
Users can sign in using credentials from social identity providers like Facebook, Google, Amazon, and Apple; or using credentials from external directories through SAML or Open ID Connect. You can manage user attribute mappings and security for federated users in your user pool.

Cognito user pool sign-in options

Choose the attributes in your user pool that are used to sign in. If you select only one attribute, or you select a user name and at least one other attribute, your user can sign in with all of the selected options. If you select only phone number and email, your user will be prompted to select one of the two sign-in options when they sign up.

User name
 Email
 Phone number

User name requirements

Allow users to sign in with a preferred user name
 Make user name case sensitive

The screenshot shows the AWS Cognito User Pools creation wizard at Step 3: Configure sign-up experience. The left sidebar lists steps 4 through 6: Step 4 (Configure message delivery), Step 5 (Integrate your app), and Step 6 (Review and create). The main content area is titled "Password policy mode" with two options: "Cognito defaults" (selected) and "Custom". Under "Cognito defaults", it specifies a password length of 8 characters and contains requirements for at least 1 number, 1 special character, 1 uppercase letter, and 1 lowercase letter. It also states that temporary passwords expire in 7 days. Below this is a section for "Multi-factor authentication" with three options: "Require MFA - Recommended" (disabled), "Optional MFA" (disabled), and "No MFA" (selected). The AWS CloudShell interface is visible at the bottom.

The screenshot shows the AWS Cognito User Pools creation wizard at Step 4: Configure sign-in experience. The left sidebar lists steps 2 through 6: Step 2 (Configure security requirements), Step 3 (Configure sign-up experience), Step 4 (Configure message delivery), Step 5 (Integrate your app), and Step 6 (Review and create). The main content area is titled "Email" and describes how user pools send email messages. It shows two options for "Email provider": "Send email with Amazon SES - Recommended" (disabled) and "Send email with Cognito" (selected). A note states that users must have configured a verified sender with Amazon SES to use the SES feature. It also specifies the SES Region as "US West (Oregon)" and the "FROM" email address as "no-reply@verificationemail.com". The "REPLY-TO" email address field is empty and labeled as optional. The AWS CloudShell interface is visible at the bottom.

The screenshot shows the AWS Cognito User Pools creation wizard at Step 3: Configure sign-up experience. The user pool name is set to "WildRydes". A warning message states: "Your user pool name can't be changed once this user pool is created." Under "Hosted authentication pages", there is an unchecked checkbox for "Use the Cognito Hosted UI". Under "Initial app client", there is a note about configuring an app client.

The screenshot shows the AWS Cognito User Pools list page. It displays one user pool named "WildRydes" created successfully. The page includes a search bar and columns for User pool name, User pool ID, Created time, and Last updated time. A "Create user pool" button is visible at the top right.

User pool name	User pool ID	Created time	Last updated time
WildRydes	us-west-2_fgj4juLB1	3 seconds ago	3 seconds ago

Copy the user pool id and client id in js/config.js in repository

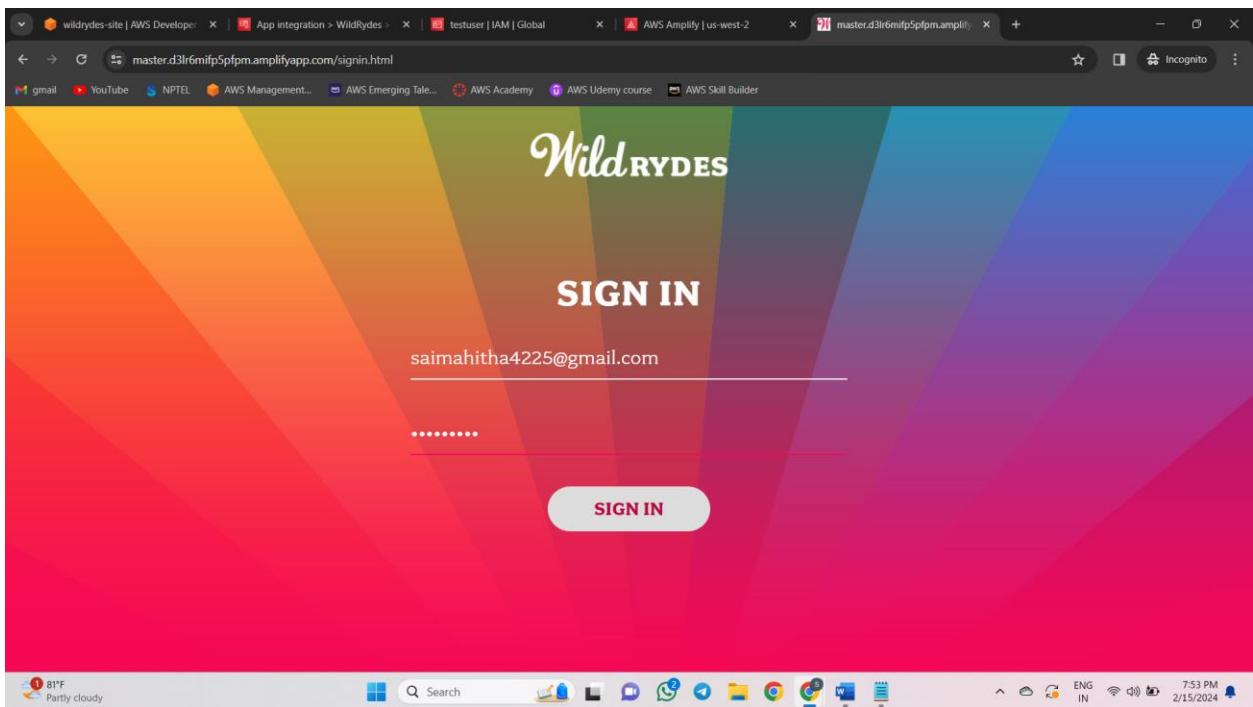
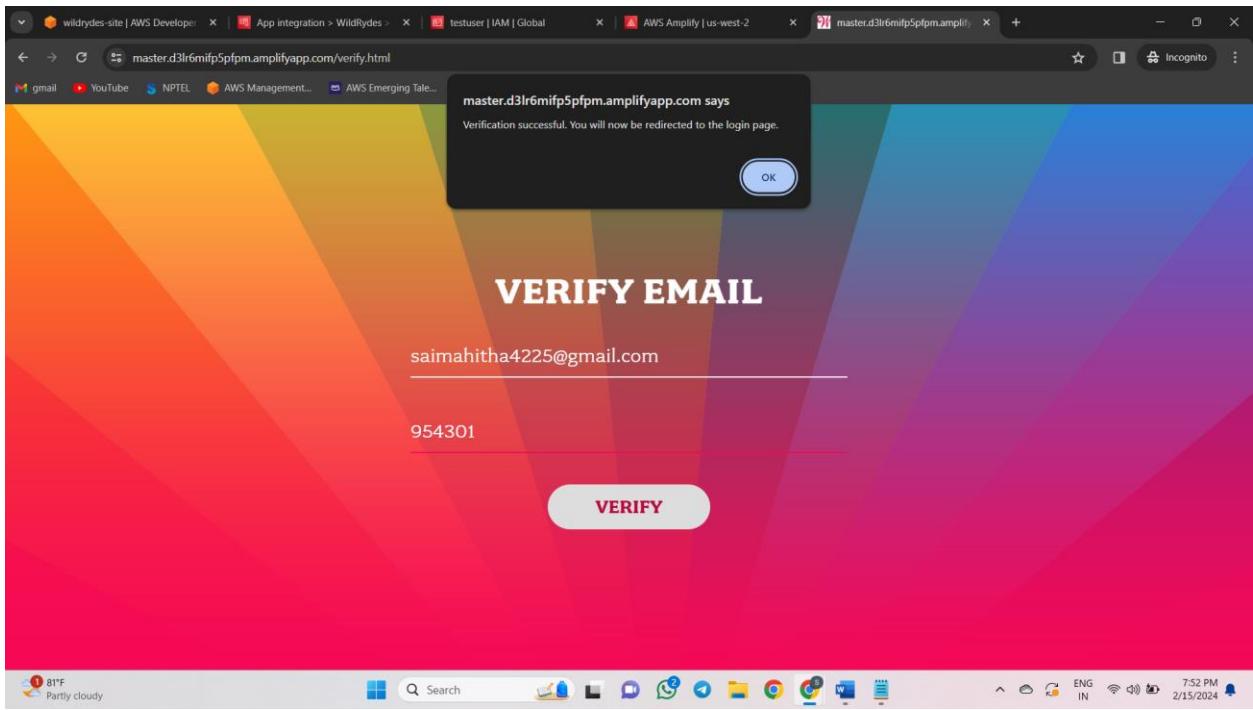
The screenshot shows the AWS CloudShell interface. On the left, there's a sidebar titled "AWS Developer Tools" with a "CodeCommit" section containing links like "Source", "Getting started", "Repositories", "Code", "Pull requests", "Commits", "Branches", "Git tags", "Settings", "Approval rule templates", "Artifacts", "Build", "Deploy", "Pipeline", and "Settings". The main area is a code editor titled "wildrydes-site / js / config.js Info". The code content is as follows:

```
1 window._config = {
2   cognito: {
3     userPoolId: 'us-west-2_fgj4juLB1', // e.g. us-east-2_0B0oG5Ab
4     userPoolClientId: '421aqfq2t1018Bvsq1hdjg53ks', // e.g. 25ddk0rj4vhfsfvruhpfi7n4hv
5     region: 'us-east-2' // e.g. us-east-2
6   },
7   api: [
8     {
9       invokeUrl: '' // e.g. https://rc7nyt4tql.execute-api.us-west-2.amazonaws.com/prod*
10    }
11 }
```

Below the code editor is a "Commit changes to master" dialog with fields for "Author name" (set to "Mahitha") and "Email address" (set to "saimahitha4225@gmail.com"). At the bottom of the screen, the Windows taskbar is visible with various icons.

Register in the web giving your email with verification code

The screenshot shows a web browser displaying the registration page for "WildRYDES". The URL in the address bar is "master.d3lr6mfp5plpm.amplifyapp.com/register.html". The page has a vibrant, multi-colored background with diagonal stripes in yellow, orange, red, green, blue, and purple. The "WildRYDES" logo is at the top center. Below it, the word "REGISTER" is prominently displayed in large, bold, white capital letters. A text input field contains the email address "saimahitha4225@gmail.com". Below the input field are two password input fields, each consisting of a series of dots. At the bottom right is a white button with the text "LET'S RYDE" in red capital letters. The browser's taskbar at the bottom shows other open tabs and system icons.



Create table in DynamoDB

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The 'Table details' step is active, where the user is defining the table name ('Rides') and partition key ('RideId'). The partition key is described as a hash value used for retrieving items and allocating data across hosts. A 'Sort key - optional' field is also present, though empty.

The screenshot shows the 'General information' section of the 'Rides' table's details page. The table has a partition key ('RideId String') and no sort key. It is in 'Provisioned' mode with an 'Active' status. Other settings shown include 'Point-in-time recovery (PITR)' off, 'DynamoDB stream' off, and 'Time to Live (TTL)' off. The ARN of the table is copied to the clipboard, and the source name (ARN) is displayed as 'arn:aws:dynamodb:us-west-2:390696895208:table/Rides'. The 'Items summary' section indicates that DynamoDB updates item counts approximately every six hours.

Create a role in IAM

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Step 1: Select trusted entity'. The title is 'Select trusted entity'. The left sidebar shows 'Step 1: Select trusted entity', 'Step 2: Add permissions', and 'Step 3: Name, review, and create'. The main content area is titled 'Trusted entity type' and contains five options:

- AWS service**: Allows AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**: Allows entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation**: Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**: Creates a custom trust policy to enable others to perform actions in this account.

Below this is a 'Use case' section with a dropdown menu set to 'Lambda'. A note says 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' At the bottom of the page, there are links for CloudShell and Feedback, and a footer with standard browser controls and a timestamp of 8:08 PM on 2/15/2024.

The screenshot shows the 'Create role' wizard in the AWS IAM console, now at Step 2: 'Add permissions'. The title is 'Add permissions'. The left sidebar shows 'Step 1: Select trusted entity', 'Step 2: Add permissions', and 'Step 3: Name, review, and create'. The main content area is titled 'Permissions policies (1/1918) Info' and shows a search bar with 'awslambdabasic' and a filter 'All types'. A table lists policies:

Policy name	Type	Description
<input checked="" type="checkbox"/> AWSLambdaBasicExecutionRole	AWS managed	Provides write permissions to CloudW...
<input type="checkbox"/> AWSLambdaBasicExecutionRole-0a3b9f7b-020f-484d...	Customer managed	-
<input type="checkbox"/> AWSLambdaBasicExecutionRole-19ef3ede-1b56-4b60...	Customer managed	-
<input type="checkbox"/> AWSLambdaBasicExecutionRole-6987b3df-9320-493f...	Customer managed	-
<input type="checkbox"/> AWSLambdaBasicExecutionRole-c161285c-5cee-4faf-b...	Customer managed	-
<input type="checkbox"/> AWSLambdaBasicExecutionRole-cb6f6808-a25a-4d6e...	Customer managed	-
<input type="checkbox"/> AWSLambdaBasicExecutionRole-d1023c57-e06e-4bf5...	Customer managed	-

At the bottom, there is a note '▶ Set permissions boundary - optional' and buttons for 'Cancel', 'Previous', and 'Next'.

Role details

Role name: WildRydesLambda

Description: Allows Lambda functions to call AWS services on your behalf.

Step 1: Select trusted entities

Trust policy:

```
1 * {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Action": [  
7                 "sts:AssumeRole"  
8             ],  
9             "Principal": {  
10                "Service": [  
11                    "lambda.amazonaws.com"  
12                ]  
13            }  
14        }  
15    ]  
16}
```

Create inline policy to that Role

Policy editor

DynamoDB

Actions allowed: putitem

Resources: table

Effect: Allow

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "dynamodb:PutItem",
      "Resource": "arn:aws:dynamodb:us-west-2:390696895208:table/Rides"
    }
  ]
}
```

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.
DynamoDBWriteAccess

Maximum 128 characters. Use alphanumeric and '+,-,_,-' characters.

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (1 of 404 services)

Service	Access level	Resource	Request condition
DynamoDB	Limited: Write	TableName string like Rides, region string like us-west-2	None

Create policy

Create a lambda function “RequestUnicorn”

RequestUnicorn

Enter a name that describes the purpose of your function.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Node.js 16.x

Architecture Info
Choose the instruction set architecture you want for your function code.
x86_64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).
 Create a new role with basic Lambda permissions
 Use an existing role
 Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
WildRydesLambda

Successfully updated the function RequestUnicorn.

Code source **Info**

Test **Deploy**

```

index.js
1 const randomBytes = require('crypto').randomBytes;
2 const AWS = require('aws-sdk');
3 const ddb = new AWS.DynamoDB.DocumentClient();
4
5 const fleet = [
6   {
7     Name: 'Angel',
8     Color: 'White',
9     Gender: 'Female',
10 },
11 {
12   Name: 'G11',
13   Color: 'White',
14   Gender: 'Male',
15 },
16 {
17   Name: 'Rocinante',
18   Color: 'Yellow',
19   Gender: 'Female',
20 },
21 ];
22
23 exports.handler = (event, context, callback) => {
24   if (!event.requestContext.authorizer) {
25     errorResponse('Authorization not configured', context.awsRequestId, callback);
26     return;
27   }
28
29   const rideId = toUrlString(randomBytes(16));
30   console.log(`Received event ('${rideId}', ${event}`);
31
32   // Because we're using a Cognito User Pools authorizer, all of the claims

```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 8:37 PM 2/15/2024

The test event TestRequestEvent was successfully saved.

Code source **Info**

Test **Deploy**

Execution result

Test Event Name		TestRequestEvent
Response	<pre> { "statusCode": 201, "body": "{\"RideId\":\"0FBeoXAZEuFmHPB-U1jnu\", \"Unicorn\":{\"Name\":\"Rocinante\", \"Color\":\"Yellow\", \"Gender\":\"Female\"}, \"Eta\":\"30 seconds\", \"Rider\":\"the_username\"}", "headers": { "Access-Control-Allow-Origin": "*" } } </pre>	
Function Logs	<pre> 2024-02-15T15:09:35.092Z 0abb7b1-e391-4ffa-bd4b-4af62da87365 Version: \$LATEST 2024-02-15T15:09:35.092Z 0abb7b1-e391-4ffa-bd4b-4af62da87365 INFO Received event (0FBeoXAZEuFmHPB-U1jnu): { path: '/ride', httpMethod: 'POST', headers: { Accept: '*/*', Authorization: 'eyJraWQlOjlLTrzN#Nz2', 'Content-Type': 'application/json; charset=UTF-8' }, queryStringParameters: null, pathParameters: null, requestContext: { authorizer: { claims: [Object] } }, body: '{ "PickupLocation":{ "Latitude":47.6174755835663, "Longitude":-122.28837066650185} }' } 2024-02-15T15:09:35.092Z 0abb7b1-e391-4ffa-bd4b-4af62da87365 INFO Finding unicorn for 47.6174755835663 , -122.28837066650185 END RequestId: 0abb7b1-e391-4ffa-bd4b-4af62da87365 Duration: 998.55 ms Billed Duration: 999 ms Memory Size: 128 MB Max Memory Used: 85 MB Init Duration: 486.33 ms REPORT RequestId: 0abb7b1-e391-4ffa-bd4b-4af62da87365 Duration: 998.55 ms Billed Duration: 999 ms Memory Size: 128 MB Max Memory Used: 85 MB Init Duration: 486.33 ms </pre>	

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 8:39 PM 2/15/2024

The screenshot shows the AWS DynamoDB 'Edit item' interface. The URL is https://us-west-2.console.aws.amazon.com/dynamodbv2/home?region=us-west-2#edit-item?itemMode=2&pk=0FBeoXAZEuFMkhP8-Uljnw&route=ROUTE_ITEM_EXPLORER&sk=&table.... The page displays an 'Attributes' table with the following data:

Attribute name	Value	Type
RideId - Partition key	0FBeoXAZEuFMkhP8-Uljnw	String
RequestTime	2024-02-15T15:09:35.001Z	String
Unicorn	Insert a field ▾	Map
Color	Yellow	String
Gender	Female	String
Name	Rocinante	String
User	the_username	String

Below the table, there are buttons for 'Form' and 'JSON view'. The status bar at the bottom shows CloudShell, Feedback, USD/EUR (-0.53%), and system icons.

Create API Gateway

The screenshot shows the AWS API Gateway 'Create REST API' interface. The URL is <https://us-west-2.console.aws.amazon.com/apigateway/main/create-rest?experience=rest®ion=us-west-2>. The page has a 'Create REST API' header and an 'API details' section with the following options:

- New API: Create a new REST API.
- Clone existing API: Create a copy of an API in this AWS account.
- Import API: Import an API from an OpenAPI definition.
- Example API: Learn about API Gateway with an example API.

Below these options, there are fields for 'API name' (WildRydes) and 'Description - optional' (empty). Under 'API endpoint type', it says 'Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.' A dropdown menu shows 'Edge-optimized' selected. At the bottom right is a 'Create API' button.

The status bar at the bottom shows CloudShell, Feedback, 80°F Haze, and system icons.

The screenshot shows the AWS API Gateway interface. On the left, a sidebar for the 'API: WildRydes' shows options like Resources, Stages, Authorizers, and Models. The main panel displays a success message: 'Successfully created REST API 'WildRydes (flgyofe2rk)''. Below this, the 'Resources' section shows a single resource with path '/'. The 'Methods (0)' section indicates 'No methods defined.' There are buttons for 'Create method', 'Update documentation', and 'Enable CORS'.

Create authorizer named “WildRydes” and test authorizer

The screenshot shows the AWS API Gateway interface. The sidebar now shows 'Authorizers' under the 'API: WildRydes' section. The main panel displays a success message: 'Successfully created REST API 'WildRydes (flgyofe2rk)''. Below this, the 'Create authorizer' section is shown. The 'Authorizer details' form has 'Authorizer name' set to 'WildRydes'. Under 'Authorizer type', 'Cognito' is selected. In the 'Cognito user pool' section, 'us-west-2' is selected from a dropdown, and 'WildRydes' is typed into a search input field. A 'Token source' field contains 'Authorization'. At the bottom, a 'Create authorizer' button is highlighted in orange.

The screenshot shows the AWS API Gateway interface for the 'WildRydes' API. On the left, a sidebar lists various API settings like Resources, Stages, and Authorizers. The 'Authorizers' section is expanded, showing 'WildRydes' with an 'Edit' and 'Delete' button. The main content area displays 'Authorizer details' for 'hbq50l'. It shows the Authorizer ID 'hbq50l', Token source 'Authorization', Cognito pool 'WildRydes - fgj4juLB1 (us-west-2)', and Token validation set to 'optional' with 'None' selected. Below this is a 'Test authorizer' section with a 'Test authorizer' button. A modal window titled 'Authorizer test: WildRydes' shows a 401 error response: 'Unauthorized request: 1f90ed39-9b0c-4bbc-9213-03d3c5d126f4'. The browser status bar at the bottom indicates it's 9:11 PM on 2/15/2024.

Create resource in API named “/ride”

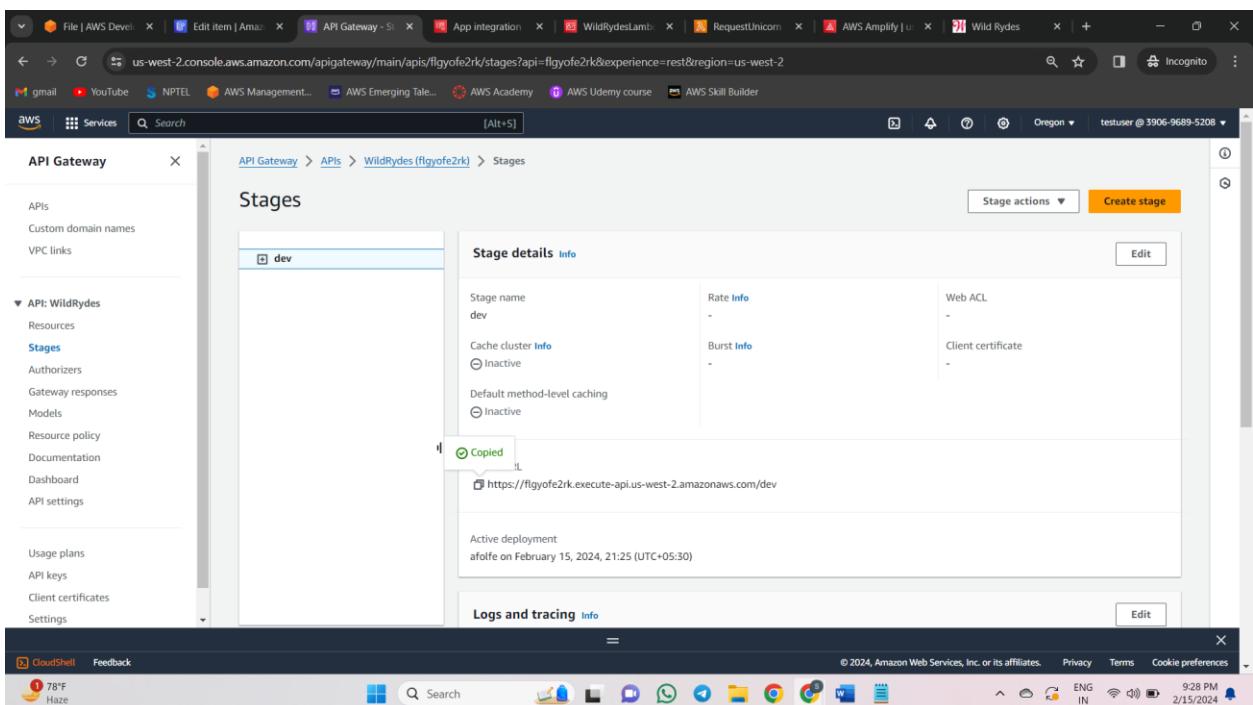
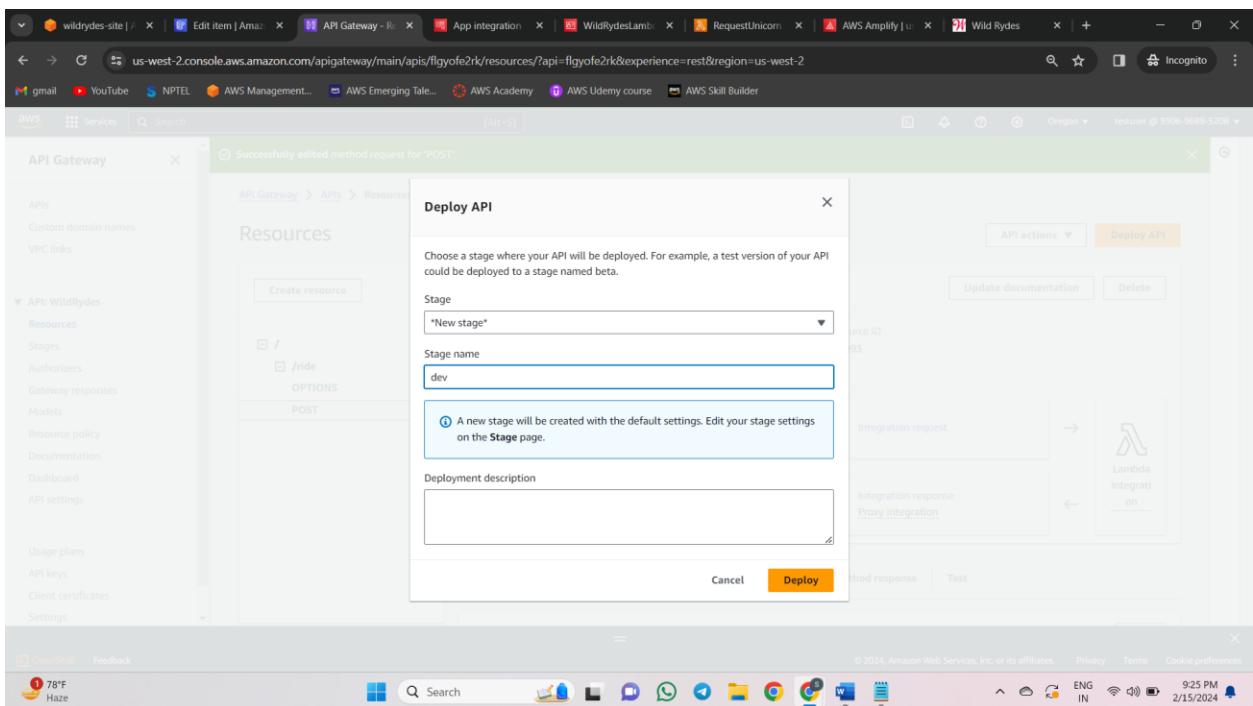
The screenshot shows the 'Create resource' page for the 'WildRydes' API. The 'Resource details' section includes a 'Proxy resource info' link, a 'Resource path' input set to '/', and a 'Resource name' input set to 'ride'. A checked checkbox for 'CORS (Cross Origin Resource Sharing) info' is present. At the bottom are 'Cancel' and 'Create resource' buttons. The browser status bar at the bottom indicates it's 9:11 PM on 2/15/2024.

Create method in resource /ride

The screenshot shows the 'Create method' dialog in the AWS Management Console. The 'Method type' is set to 'POST'. Under 'Integration type', 'Lambda proxy integration' is selected, indicated by a blue outline. Other options like 'HTTP', 'Mock', 'AWS service', and 'VPC link' are shown with their respective icons. Below this, a dropdown for 'Lambda function' shows 'us-west-2' and a search bar containing 'arn:aws:lambda:us-west-2:390696895208:function:Req'. A note at the bottom says 'Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.' The status bar at the bottom indicates 'CloudShell Feedback' and shows system information.

Edit the method request and deploy API

The screenshot shows the 'Edit method request' dialog. Under 'Method request settings', the 'Authorization' dropdown is set to 'WildRydes'. The 'Request validator' dropdown is set to 'None'. The 'Operation name - optional' field contains 'GetPets'. A section for 'URL query string parameters' is visible below. The status bar at the bottom indicates 'CloudShell Feedback' and shows system information.



Copy the invoke url in /js/config.js, edit ride.html and check deployment in Amplify

wildrydes-site / js / config.js Info

The code editor uses the Tab key to control indentation. To navigate away from the code editor, use Escape plus Tab keys.

```
1 window._config = {
2   cognito: {
3     userPoolId: 'us-west-2_fgj4juBL1', // e.g. us-east-2_u00og5pAb
4     userPoolClientId: '42laqHg2tI0isovsqihdgjg5ks', // e.g. 25ddkanj4v6hfsFvrupf17n4hv
5     region: 'us-east-2' // e.g. us-east-2
6   },
7   api: {
8     invokeUrl: 'https://flyyofe2rk.execute-api.us-west-2.amazonaws.com/dev' // e.g. https://rc7nyt4tql.execute-api.us-west-2.amazonaws.com/prod',
9   }
10 };
11
```

Commit changes to master

File: wildrydes-site/js/config.js

Author name

Email address

AWS CloudShell

File | AWS Dev | Edit item | Amazon CloudWatch Metrics | API Gateway - Staging | App integration | WildrydesLambda | RequestUnicorn | AWS Amplify | Wild Rydes | + Incognito

us-west-2.console.aws.amazon.com/codesuite/codecommit/repositories/wildrydes-site/files/edit/refs/heads/master/-/ride.html?region=us-west-2

gmail YouTube S NPTEL AWS Management... AWS Emerging Tale... AWS Academy AWS Udemy course AWS Skill Builder

AWS Services Search [Alt+S]

Developer Tools CodeCommit

Source • CodeCommit

Getting started

Repositories

Code

Pull requests

Commits

Branches

Git tags

Settings

Approval rule templates

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Settings

Developer Tools > CodeCommit > Repositories > wildrydes-site > File

Edit a file

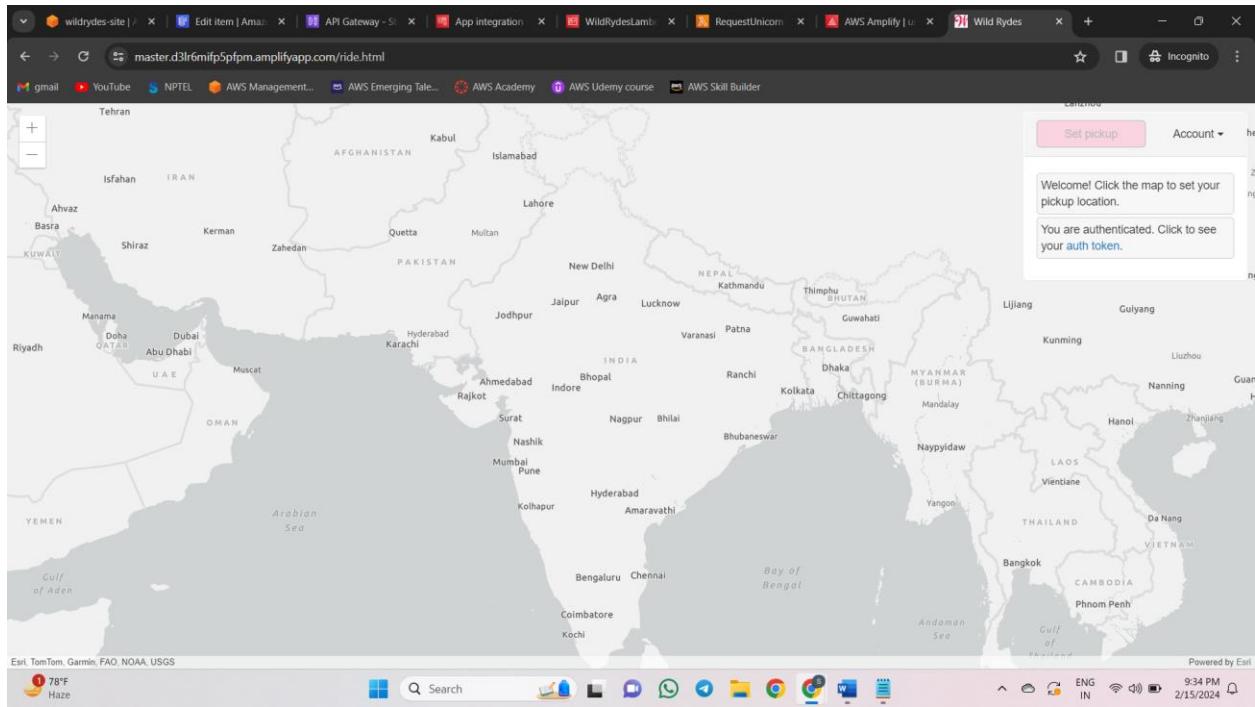
Reference master

wildrydes-site / ride.html Info

The code editor uses the Tab key to control indentation. To navigate away from the code editor, use Escape plus Tab keys.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no">
6     <title>Wild Rydes</title>
7     <meta charset="utf-8">
8     <meta http-equiv="X-UA-Compatible" content="IE=edge">
9     <meta name="viewport" content="width=device-width, initial-scale=1">
10    <meta name="description" content="Serverless web application example">
11    <meta name="author" content="">
12
13
14
15    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeKdGmRAkuyuHHL5tLVn7fkb46vSbE9WUOYjbaQq+8tTfY9glcF9" crossorigin="anonymous">
16    <link rel="stylesheet" href="https://js.arcgis.com/4.6/esri/css/main.css">
17    <link rel="stylesheet" href="/css/ride.css">
18    <link rel="stylesheet" href="/css/message.css">
19  </head>
20
21  <body>
22
23    <div class="info panel panel-default">
24      <div class="panel-heading">
```

OUTPUT:



Requesting a unicorn at some point

